CYCLOSTATIONARY FEATURE SELECTION

Anthony Tai, Ph.D., and Savannah Farney
Spectrum Warfare Department
Naval Surface Warfare Center - Crane Division
Crane, IN 47522-5001
{anthony.s.tai.civ, savannah.l.farney.civ}@us.navy.mil

ABSTRACT

This article describes an effective procedure for identifying cyclostationary features that strongly influence the performance of certain standard machine learning prediction models. Cyclostationary features such as moments and cumulants extracted from raw electromagnetic signals are often utilized in detecting and classifying digitally modulated signals. However, not all features contribute equally to the outcome. In this work, we explore and implement SHAP (SHapley Additive exPlanations), a game theoretic method to determine the significance of each cyclostationary feature in predicting the modulation types associated with received signals. Using properly reduced feature sets obtained from SHAP, we demonstrate that models such as XGBoost and Random Forest could achieve classification accuracy comparable to the baseline with full feature sets. Furthermore, our empirical results indicate that a balanced choice of significant features could improve computational efficiency without compromising prediction performance. Using cyclostationary feature selection as a use case, we show that the suggested approach could be applied to a broader range of datasets and machine learning techniques to identify and quantitatively explain the factors that most likely influence prediction model results.

1 INTRODUCTION

Advancements in artificial intelligence (AI) and machine learning (ML) have produced powerful systems that contributed to many applications across scientific domains such as healthcare, finance, automobile, criminal investigation, and military operations. These applications were made possible with the continuous development in computation algorithms, prediction model architecture, simulated and real-world system training. However, while an AI/ML model might produce results as expected, its operations may not be totally transparent or understood.

In order to raise our confidence in using the AI/ML technology to complete important tasks, it is imperative to comprehend and trace back to how models arrive at their outcomes. In addition to assessing their accuracy and computational costs, we also measure the consistency in their performance. Since the predicted results rely on the input data, the model structure, the task it aims to accomplish, establishing an explainable relationship linking the data to the model is critical.

Lloyd Shapley introduced the Shapley value as a solution concept in cooperative game theory. Given the set of factors (or features) that produces a certain prediction, the Shapley value method computes for each factor's contribution to the prediction outcome. Based on the principle of Shapley value, Lundberg and Lee (Lundberg and Lee, 2017) developed a unified framework, SHapley Additive exPlanations or SHAP, for interpreting machine learning predictions.

In this article, we first discuss the approach and methodology for testing and evaluating the SHAP algorithm. This is followed by a description of the datasets selected and the experiments designed for model assessment. Then, we present our empirical results and some insights. Lastly, we conclude with our plans for future research.

2 APPROACH AND METHODOLOGY

We first searched for and investigated documentation and existing Python code for SHAP. The SHAP installation procedure is described online in detail at `https://shap.readthedocs.io/en/latest/index.html`. The weblog (blog) also provides example use cases on linear models, tree-based models, and neural networks, with applications in sentiment analysis, text translation, and question answering.

Then, we implemented the algorithm on a linear regression model to duplicate the results shown in the blog. Once code validation was completed, as described in § 2.1, we prepared model training and test sets consisting of cyclostationary feature values.

Cyclostationary signal processing has many applications in mobile communication, radar, and sonar (Napolitano, 2013). In particular, cyclostationary features extracted from raw electromagnetic waveforms are useful for modulation recognition in signal classification tasks (Spooner et al., 2017; O'Shea et al., 2018). The description of the cyclostationary feature datasets used in our experiments is provided in § 2.2

2.1 SHAP IMPLEMENTATION - A SIMPLE LINEAR REGRESSION EXAMPLE

Following the procedure, we applied SHAP on a classic housing price dataset `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html`. The dataset consists of 20,640 blocks of houses across California in 1990 and 9 features: 'MedInc', 'HouseAge', 'AveRooms', 'AveBedrms', 'Population', 'AveOccup', 'Latitude', 'Longitude', 'Price', defined as:

- MedInc - median income in block group
- HouseAge - median house age in block group
- AveRooms - average number of rooms per household
- AveBedrms - average number of bedrooms per household
- Population - block group population
- AveOccup - average number of household members
- Latitude - block group latitude
- Longitude - block group longitude
- Price - housing price

We computed the coefficients of a linear additive regression model which quantified the changes in the model output in accordance with the changes in individual features, as shown in Table 1 below.

Table 1: Model coefficients of features in the California Housing Price dataset.

Housing Price Feature	Model Coefficient Value
MedInc	0.42563
HouseAge	0.01033
AveRooms	-0.1161
AveBedrms	0.66385
Population	3e-05
AveOccup	-0.26096
Latitude	-0.46734
Longitude	-0.46272

Because of scaling effects, measuring the magnitude alone will not tell us the absolute impact of each feature. However, with a classical partial dependence plot, we can visualize the distribution of a particular feature in relation to the model output. The partial dependence plots pertaining to the Housing Price feature MedInc are depicted in the left panels in Figure 1.

As discussed in the blog https://shap.readthedocs.io/en/latest/example_notebooks/overviews/An%20introduction%20to%20explainable%20AI%20with%20Shapley%20values.html#nlp_model, computing SHAP values is difficult. On the other hand, for a linear model the SHAP value can be obtained by taking the difference between the expected model output and the partial dependence plot at the feature's value. Examples are shown in the center panel.

As for Shapley values, the contributions of individual features to the outcome of a trained prediction model can be easily visualized using a waterfall plot. This result hinges on the fact that the sum of the features always equals the difference between the expected and the current model output. The waterfall plot for a linear model is given in the upper right panel. It shows how we get from the baseline values $E[f(x)]$ = 1.904 to the predicted value $f(x)$ = 1.681. The top three contributors are:

- Latitude: 32.57
- Longitude = -117.07
- MedInce = 2.586

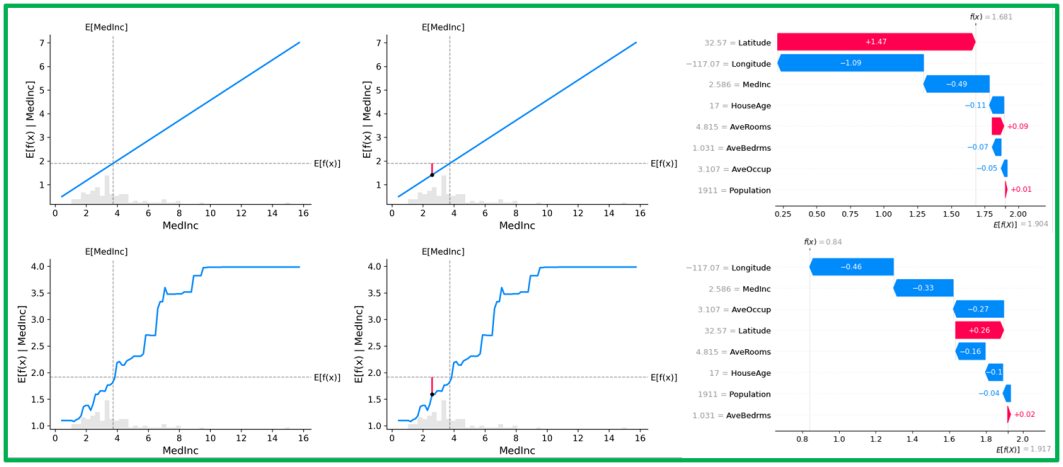

Figure 1: Informational plots pertaining to the Housing Price feature MedInc. Left panel: classical partial dependence plots. Center panel: SHAP values from partial dependence plots. Right panel: waterfall plots. Plots in the upper half were obtained from applying SHAP on a simple linear regression model. Plots in the lower panel were obtained from using a generalized additive model.

2.2 Cyclostationary Signals

Cyclostationary signals are signals with periodic statistical parameters such as their mean values, variances, and moments (Spooner, Chad. "Understanding and Using the Statistics of Communication Signals." Cyclostationary Signal Processing, 2022. https://cyclostationary.blog/2015/09/28/welcome-to-the-csp-blog/). As an example, we can consider the autocorrelation function of a digitally modulated signal x: $R_x(\tau) = E[x(t+\tau)x(t)]$. If the signal x is wide-sense stationary, that is, $x(t) = x(t+T_0)\ \forall t \in \mathbb{R}$, then its autocorrelation function is periodic with period of cyclostationarity T_0: $R_x(t+T_0, \tau) = R_x(t, \tau)\ \forall t \in \mathbb{R}$ (Napolitano, 2014). Since the autocorrelation function is periodic, it can be expressed in Fourier series expansion:

$$R_x(t,\tau) = \sum_{n=-\infty}^{+\infty} R_x^{n/T_0}(\tau) e^{j2\pi(\frac{n}{T_0})t}, \text{ cycle frequencies: } \frac{n}{T_0}, n \in \mathbb{Z}.$$

Correspondingly, the Cyclic Autocorrelation Function (CAF) is defined as

$$R_x^{n/T_0}(\tau) = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} R_x(t,\tau) e^{-j2\pi(\frac{n}{T_0})t}\, dt,$$

and the Spectral Correlation Function (SCF) is given by
$S_x^{n/T_0}(f) = \int_R R_x^{n/T_0}(\tau) e^{-j2\pi f\tau}\, d\tau.$

On the other hand, higher order statistics including the temporal and spectral moments and cumulants are described in detail in (Spooner and Gardner, 1994). A plot of cyclostationary moments and cumulants associated with various modulation types is given in Figure 2.

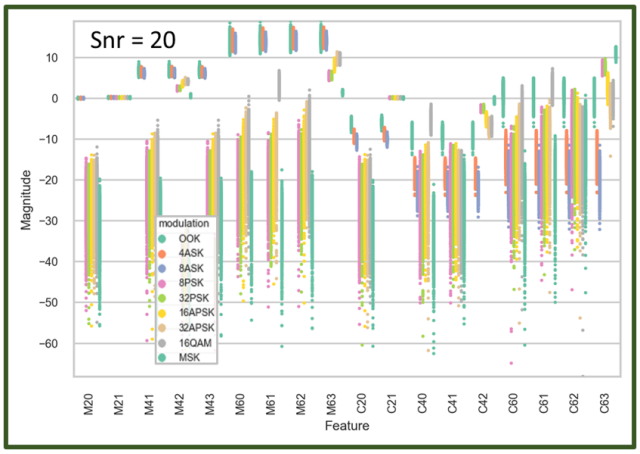

Figure 2: Cyclostationary features associated with various modulation types. Mnm denotes moment of order n with m conjugations, Cnm denotes cumulant of order n with m conjugations, and SNR stands for signal-to-noise ratio.

2.3 Datasets

A collection of seven datasets were employed in our experiments described in the next section. These synthetic data sets were generated to represent the magnitudes of eighteen cyclostationary features extracted from raw electromagnetic signals with nine different digital modulation types. Specifically, the modulations included OOK, MSK, 4ASK, 8ASK, 8PSK, 32PSK, 16APSK, 32APSK, and 16QAM. Two types of cyclostationary features were employed: the group of Moments consisted of M20, M21, M41, M42, M43, M60, M61, M62, and M63, while the group of Cumulants was composed of C20, C21, C40, C41, C42, C60, C61, C62, and C63. Furthermore, in order to investigate the effects of noise on SHAP values, we also considered seven signal-to-noise (SNR) values $\in \{0, \pm 5. \pm 10, 15, 20\}$ in generating the input data.

2.4 Experiments and Results

We conducted our experiments on two popular machine learning (ML) models, XGBoost and Random Forest, in three phases. First, we established a baseline for each prediction model by performing waveform classification on the entire set of cyclostationary features over the seven SNR values. In the second phase, we applied the SHAP method to determine the importance of each feature. Based on the results we ranked the features and set the criteria for top feature selection. Finally, we repeated the process of waveform classification on the reduced set of selected features. The entire process was executed for both XGBoost and Random Forest models.

As a demonstration, the empirical results obtained with XGBoost models for the case SNR = 20 are given below. Results for other scenarios are provided in the appendix.

Computing and Ranking by Importance
Utilizing the SHAP algorithm, for each cyclostationary feature we obtained not only its Importance value, therefore its contribution to the model output, but also the proportion of presence of each modulation type. For example, the most influential feature for the present configuration is the Cumulant C42 with an Importance value of 0.98, as shown on the left hand side in Figure 3. The significance of a particular modulation type is indicated by its length of proportion.

SHAP Values

	Features	Importance
4	C42	0.982897
0	C20	0.580637
15	M61	0.343460
2	C40	0.336978
8	C63	0.280748
17	M63	0.263459
1	C21	0.204932
3	C41	0.159347
12	M42	0.105930
14	M60	0.084225
5	C60	0.069728
9	M20	0.060988
6	C61	0.044370
7	C62	0.044286
16	M62	0.043492
11	M41	0.043311
10	M21	0.036777
13	M43	0.000000

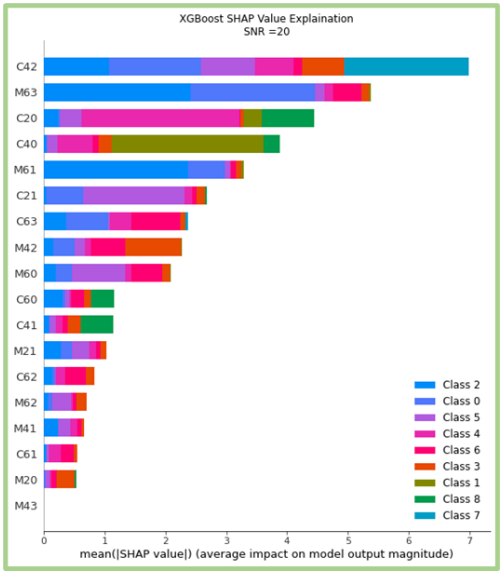

Figure 3: Ranking cyclostationary features by Importance. Left panel: SHAP value table showing Importance value for each feature. Right panel: bar-chart depicting significant modulation types.

Examining XGBoost prediction accuracy with confusion matrices

To qualitatively measure the impact of significant and insignificant cyclostationary features in classifying modulated signals, we compared the model prediction accuracy resulting from reduced feature sets to the baseline where the full feature set was used as the input to the model. Our experiments showed that at SNR=20, XGBoost maintained a healthy performance even when eleven least important features were removed from the datasets. This is shown in Figure 4

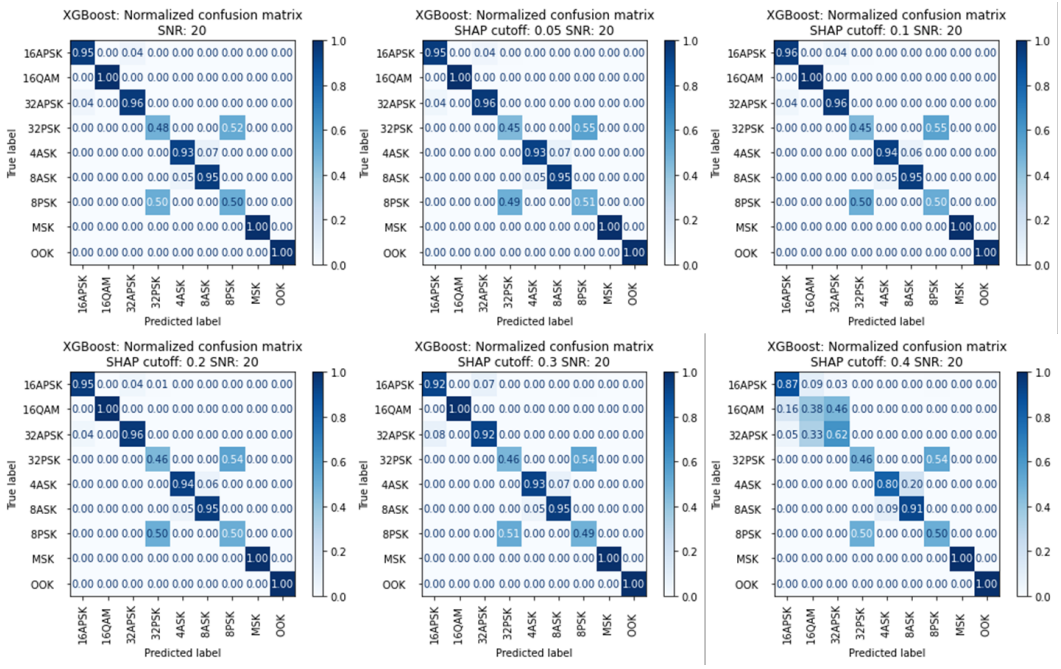

Figure 4: Examining XGBoost prediction accuracy with confusion matrices. The baseline is given in the upper left-hand corner.

The overall XGBoost prediction Accuracy on reduced cyclostationary feature sets is presented in Figure 5

XGBoost SNR=20 OS=2				
Lowest SHAP value of Remaining Variables	Variables removed	Accuracy percent with all Variables	Accuracy percentage After Using SHAP to Remove Predictors	Percentage Lost
.05	6	86.389	86.111	0.278
.1	9		86.361	0.028
0.2	11		86.306	0.083
0.3	14		85.250	1.139
0.4	16		72.778	13.611

Figure 5: Overall XGBoost prediction Accuracy on reduced cyclostationary feature set

Model accuracy vs. computational time
Finally, we also investigated the trade-off between model prediction accuracy and computational time associated with different Cut-off threshold values. It is important to note that these two metrics do not necessarily move in the opposite direction. For instance, in Figure 6 we observe that while XGBoost continued to maintain high accuracy until 14 features were removed from the dataset, its computational time dropped noticeably when 9 or more features were taken out. The right-most plot in the figure suggests that excluding the 14 least significant cyclostationary features from the dataset could produce an optimal trade-off between model accuracy and processing time.

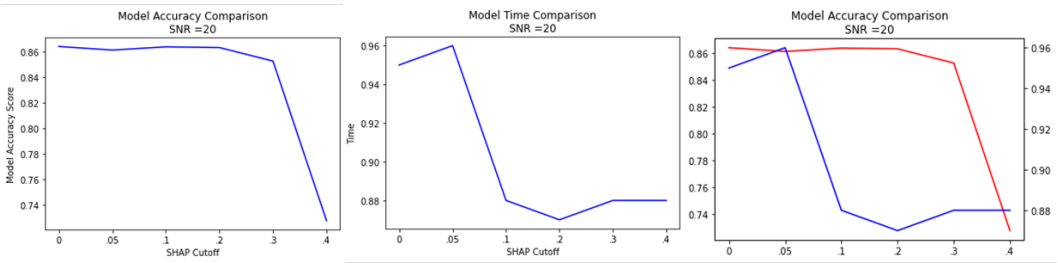

Figure 6: Model accuracy vs. computational time. Excluding the 14 least significant cyclostationary features from the dataset could produce an optimal trade-off between model accuracy and processing time.

3 Conclusions and Future Work

In this work, we explored and implemented SHAP (SHapley Additive exPlanations), a game theoretic method to determine the significance of each cyclostationary feature in predicting the modulation types associated with received signals. We demonstrated that models such as XGBoost and Random Forest could achieve classification accuracy comparable to the baseline with full feature sets. Finally, our empirical results indicated that an appropriate choice of significant features could improve computational efficiency without compromising prediction performance. Our hope is that the suggested approach could be applied to a broader range of datasets and machine learning techniques to identify and quantitatively explain the factors that most likely influence prediction model results.

Our future research will seek to accomplish two main tasks. First, we want to assess the effectiveness of SHAP on the output of SLQ-32 shipboard electronic warfare system. In addition, we want to study if deep neural network architectures such as convolutional neural networks and residual nets could benefit from applying the SHAP method.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support provided by the 2023 Naval Innovative Science & Engineering (NISE) program at the Naval Surface Warfare Center - Crane Division.

REFERENCES

Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Antonio Napolitano. Generalizations of Cyclostationarity: A New Paradigm for Signal Processing for Mobile Communications, Radar, and Sonar. *IEEE Signal Processing Magazine*, 30(6):53–63, November 2013. ISSN 1558-0792. doi: 10.1109/MSP.2013.2265101. Conference Name: IEEE Signal Processing Magazine.

Antonio Napolitano. Cyclostationary Signal Processing and its Generalizations, 2014. URL https://drive.google.com/file/d/1AWk__SK8_hFFD4ERvWIGyXKj6qNd6SNx/view?usp=sharing&usp=embed_facebook.

Timothy James O'Shea, Tamoghna Roy, and T. Charles Clancy. Over-the-Air Deep Learning Based Radio Signal Classification. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):168–179, February 2018. ISSN 1941-0484. doi: 10.1109/JSTSP.2018.2797022. Conference Name: IEEE Journal of Selected Topics in Signal Processing.

Chad M. Spooner, Apurva N. Mody, Jack Chuang, and Josh Petersen. Modulation recognition using second- and higher-order cyclostationarity. In *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 1–3, March 2017. doi: 10.1109/DySPAN.2017.7920744.

C.M. Spooner and W.A. Gardner. The cumulant theory of cyclostationary time-series. II. Development and applications. *IEEE Transactions on Signal Processing*, 42(12):3409–3429, December 1994. ISSN 1941-0476. doi: 10.1109/78.340776. Conference Name: IEEE Transactions on Signal Processing.

Appendices

Additional Empirical Results are provided below for

- Random Forest Model at SNR = 20
- XGBoost Model at SNR = 15
- Random Forest Model at SNR = 15
- XGBoost Model at SNR = 10
- Random Forest Model at SNR = 10

A RANDOM FOREST MODEL AT SNR = 20

SHAP Values

	Features	Importance
17	M63	0.406083
12	M42	0.311270
4	C42	0.306166
8	C63	0.302265
1	C21	0.179598
15	M61	0.177706
16	M62	0.143738
0	C20	0.135937
14	M60	0.128155
13	M43	0.125639
9	M20	0.118764
11	M41	0.103892
2	C40	0.091192
6	C61	0.082254
3	C41	0.075376
5	C60	0.058203
7	C62	0.051958
10	M21	0.028060

[No Title]

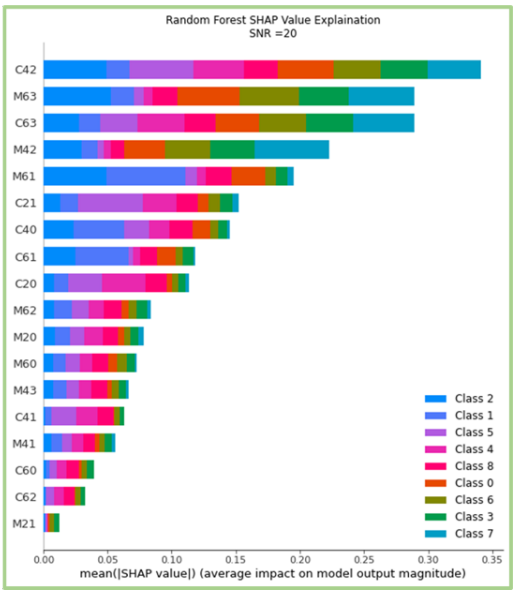

Figure 7: Random Forest Model at SNR = 20. Ranking cyclostationary features by Importance. Left panel: SHAP value table showing Importance value for each feature. Right panel: bar-chart depicting significant modulation types.

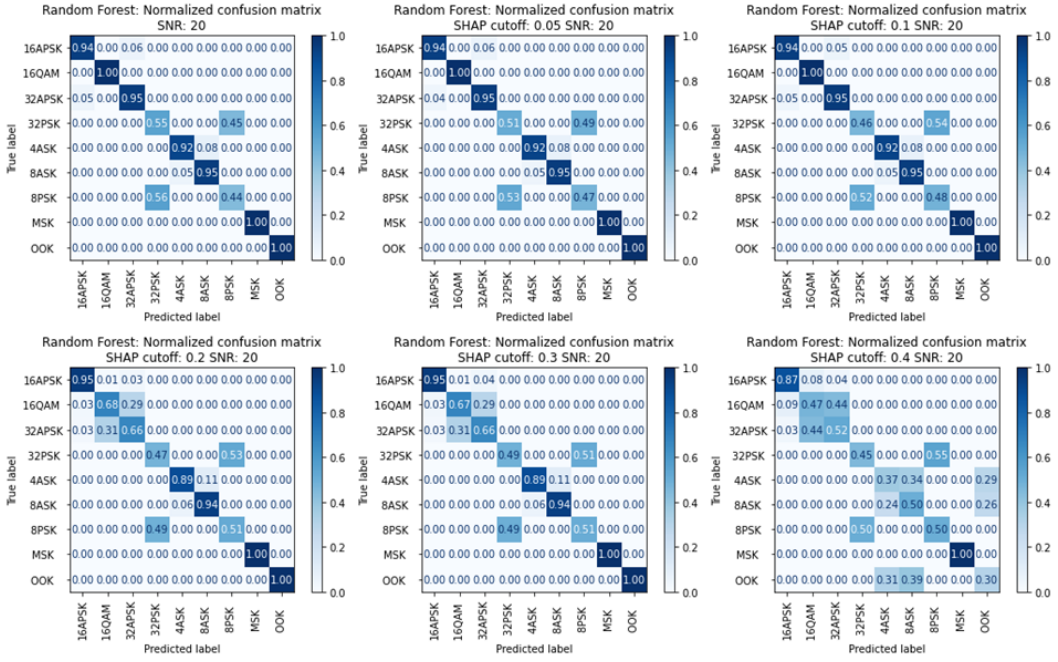

Figure 8: Random Forest Model at SNR = 20. Examining Random Forest prediction accuracy with confusion matrices. The baseline is given in the upper left-hand corner.

Random Forest SNR=20 OS=2				
Lowest SHAP value of Remaining Variables	Variables removed	Accuracy percent with all Variables	Accuracy percentage After Using SHAP to Remove Predictors	Percentage Lost
.05	1	86.125	86.153	-0.028
.1	6		85.778	0.347
0.2	14		79.000	7.125
0.3	14		79.014	7.111
0.4	17		55.236	30.889

Figure 9: Random Forest Model at SNR = 20. Overall Random Forest prediction Accuracy on reduced cyclostationary feature set

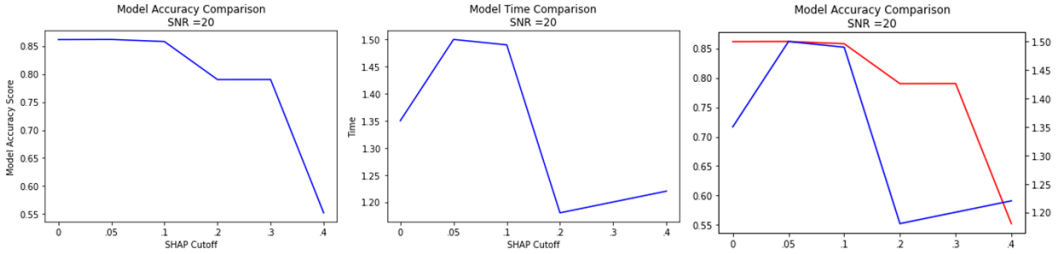

Figure 10: Random Forest Model at SNR = 20. Model accuracy vs. computational time. Excluding the 14 least significant cyclostationary features from the dataset could produce an optimal trade-off between model accuracy and processing time.

B XGBoost Model at SNR = 15

SHAP Values

	Features	Importance
4	C42	1.165529
0	C20	0.486193
2	C40	0.353258
17	M63	0.293268
3	C41	0.285587
15	M61	0.215666
1	C21	0.208926
8	C63	0.141968
12	M42	0.136634
16	M62	0.101029
9	M20	0.088748
14	M60	0.064140
5	C60	0.062932
7	C62	0.057575
6	C61	0.057330
11	M41	0.049005
10	M21	0.026842
13	M43	0.000000

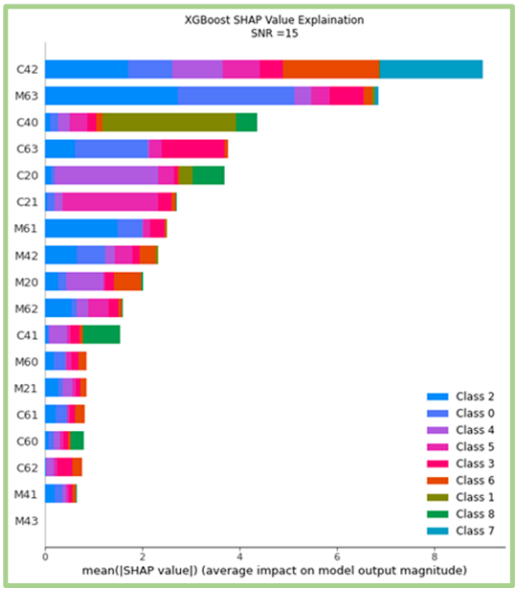

Figure 11: XGBoost Model at SNR = 15. Ranking cyclostationary features by Importance. Left panel: SHAP value table showing Importance value for each feature. Right panel: bar-chart depicting significant modulation types.

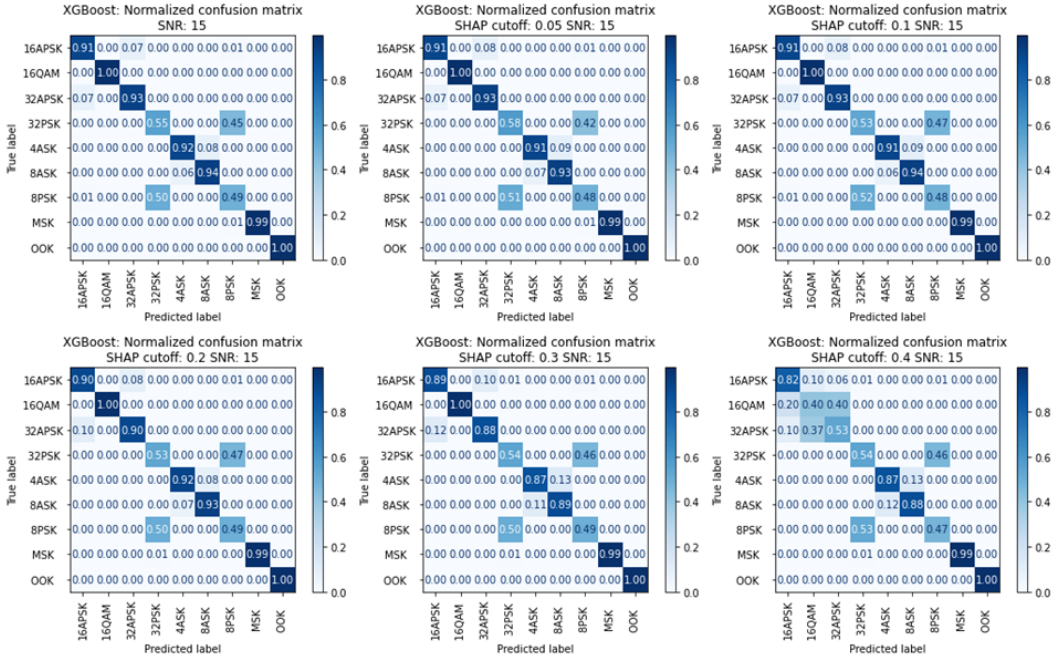

Figure 12: XGBoost Model at SNR = 15. Examining XGBoost prediction accuracy with confusion matrices. The baseline is given in the upper left-hand corner.

XGBoost SNR=15 OS=2				
Lowest SHAP value of Remaining Variables	Variables removed	Accuracy percent with all Variables	Accuracy percentage After Using SHAP to Remove Predictors	Percentage Lost
.05	3	85.889	86.028	-0.139
.1	8		85.389	0.500
0.2	11		85.278	0.611
0.3	15		83.861	2.028
0.4	16		72.292	13.597

Figure 13: XGBoost Model at SNR = 15. Overall XGBoost prediction Accuracy on reduced cyclostationary feature set

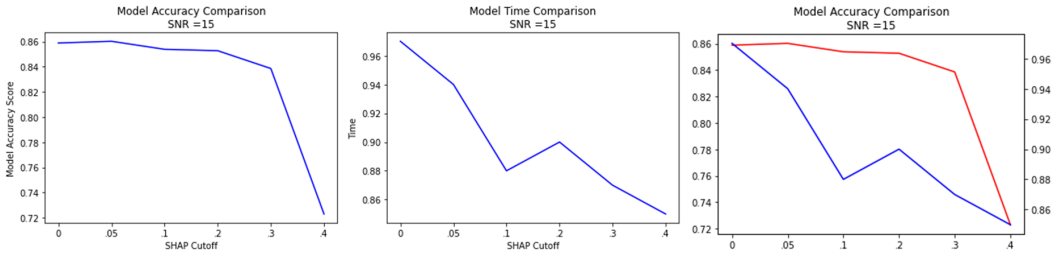

Figure 14: XGBoost Model at SNR = 15. Model accuracy vs. computational time. Excluding the 15 least significant cyclostationary features from the dataset could produce an optimal trade-off between model accuracy and processing time.

C RANDOM FOREST MODEL AT SNR = 15

SHAP Values

	Features	Importance
8	C63	0.543461
4	C42	0.395986
17	M63	0.242711
12	M42	0.226111
9	M20	0.210049
14	M60	0.180398
13	M43	0.171893
16	M62	0.169685
15	M61	0.168582
1	C21	0.164929
11	M41	0.147975
0	C20	0.110573
2	C40	0.076204
6	C61	0.071973
7	C62	0.059483
3	C41	0.058156
5	C60	0.057006
10	M21	0.027246

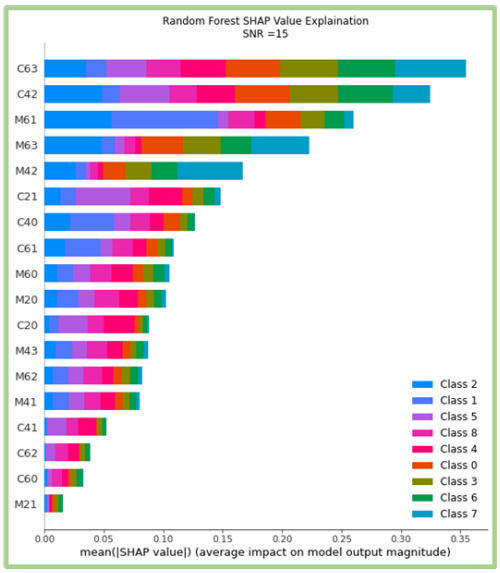

Figure 15: Random Forest Model at SNR = 15. Ranking cyclostationary features by Importance. Left panel: SHAP value table showing Importance value for each feature. Right panel: bar-chart depicting significant modulation types.

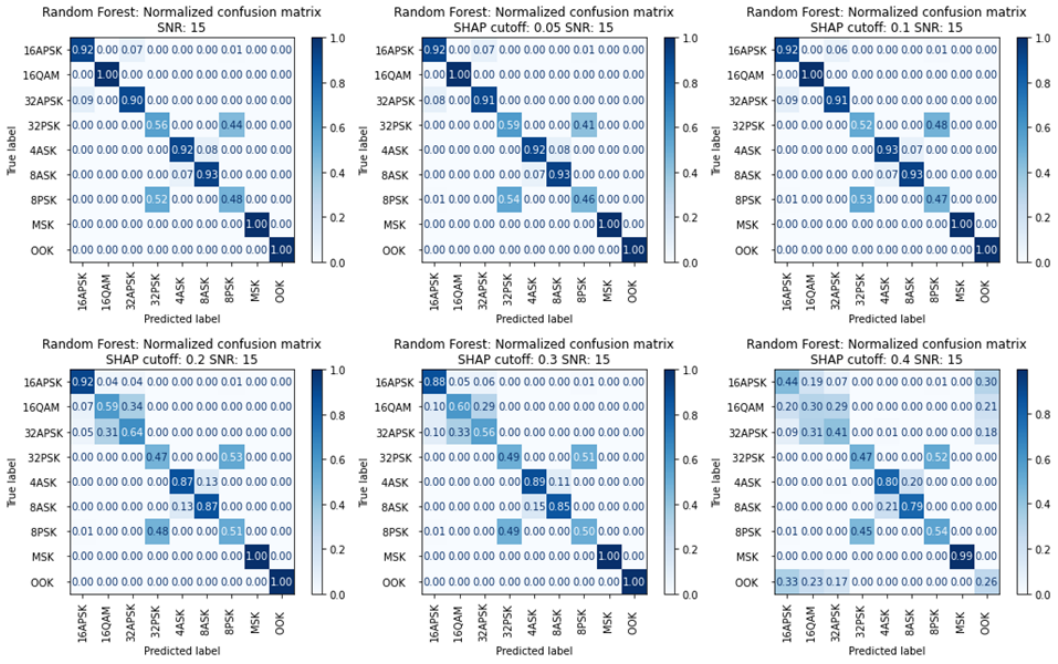

Figure 16: Random Forest Model at SNR = 15. Examining Random Fores prediction accuracy with confusion matrices. The baseline is given in the upper left-hand corner.

Random Forest SNR=15 OS=2				
Lowest SHAP value of Remaining Variables	Variables removed	Accuracy percent with all Variables	Accuracy percentage After Using SHAP to Remove Predictors	Percentage Lost
.05	1	85.639	85.847	-0.208
.1	6		85.417	0.222
0.2	13		76.444	9.194
0.3	16		75.264	10.375
0.4	17		55.097	30.542

Figure 17: Random Forest Model at SNR = 15. Overall Random Fores prediction Accuracy on reduced cyclostationary feature set

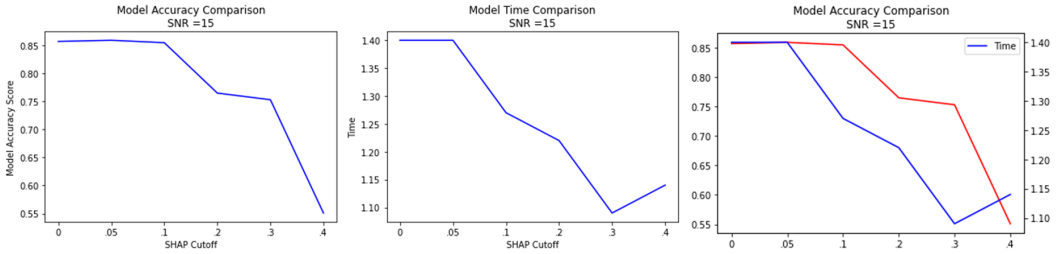

Figure 18: Random Forest Model at SNR = 15. Model accuracy vs. computational time. Excluding the 6 least significant cyclostationary features from the dataset could produce an optimal trade-off between model accuracy and processing time.

D XGBoost Model at SNR = 10

SHAP Values

	Features	Importance
8	C63	0.964257
0	C20	0.491907
17	M63	0.360257
4	C42	0.340315
3	C41	0.312446
15	M61	0.225394
14	M60	0.189372
2	C40	0.186928
1	C21	0.179092
12	M42	0.124868
7	C62	0.097299
9	M20	0.093930
6	C61	0.083173
16	M62	0.081630
11	M41	0.079507
10	M21	0.068230
5	C60	0.048522
13	M43	0.000000

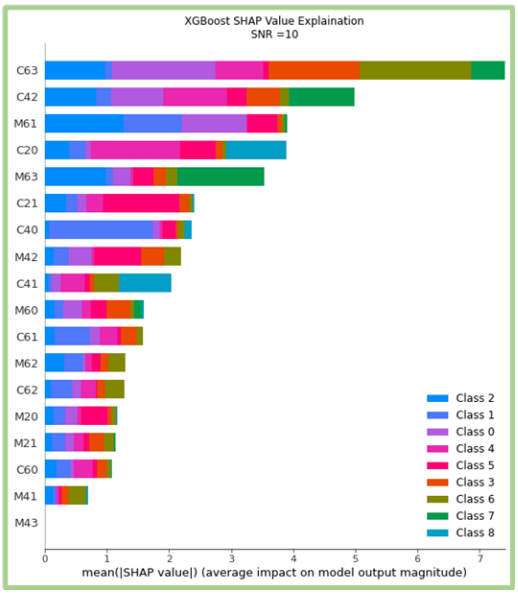

Figure 19: XGBoost Model at SNR = 10. Ranking cyclostationary features by Importance. Left panel: SHAP value table showing Importance value for each feature. Right panel: bar-chart depicting significant modulation types.

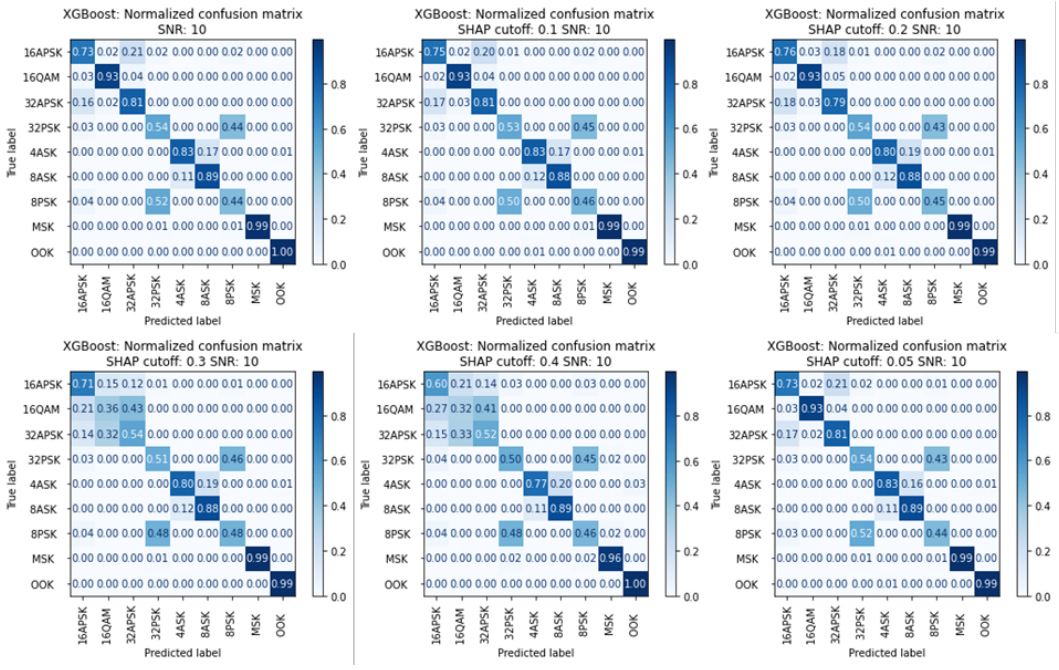

Figure 20: XGBoost Model at SNR = 10. Examining XGBoost prediction accuracy with confusion matrices. The baseline is given in the upper left-hand corner.

XGBoost SNR=10 OS=2				
Lowest SHAP value of Remaining Variables	Variables removed	Accuracy percent with all Variables	Accuracy percentage After Using SHAP to Remove Predictors	Percentage Lost
.05	2	79. 472	79.556	-0.083
.1	8		79.583	-0.111
0.2	12		79.181	0.292
0.3	13		69.389	10.083
0.4	16		66.694	12.778

Figure 21: XGBoost Model at SNR = 10. Overall XGBoost prediction Accuracy on reduced cyclostationary feature set

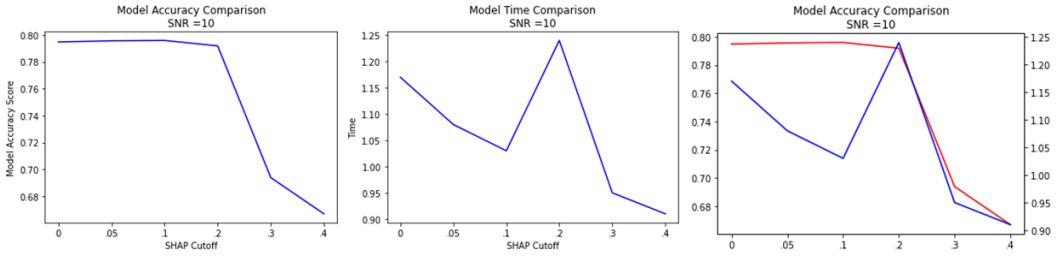

Figure 22: XGBoost Model at SNR = 10. Model accuracy vs. computational time. Excluding the 8 least significant cyclostationary features from the dataset could produce an optimal trade-off between model accuracy and processing time.

E RANDOM FOREST MODEL AT SNR = 10

SHAP Values

	Features	Importance
8	C63	0.863046
14	M60	0.361490
4	C42	0.290377
13	M43	0.264043
9	M20	0.232809
15	M61	0.195256
17	M63	0.166311
1	C21	0.162186
11	M41	0.161475
16	M62	0.132382
12	M42	0.107834
2	C40	0.105634
0	C20	0.105122
7	C62	0.083495
6	C61	0.074356
3	C41	0.072768
5	C60	0.058953
10	M21	0.035278

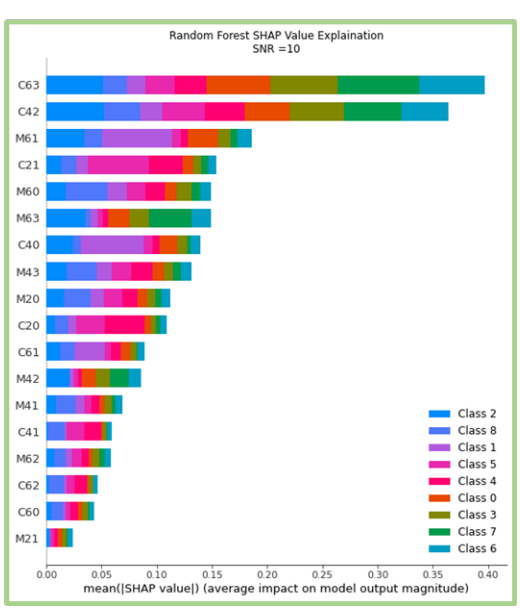

Figure 23: Random Forest Model at SNR = 10. Ranking cyclostationary features by Importance. Left panel: SHAP value table showing Importance value for each feature. Right panel: bar-chart depicting significant modulation types.

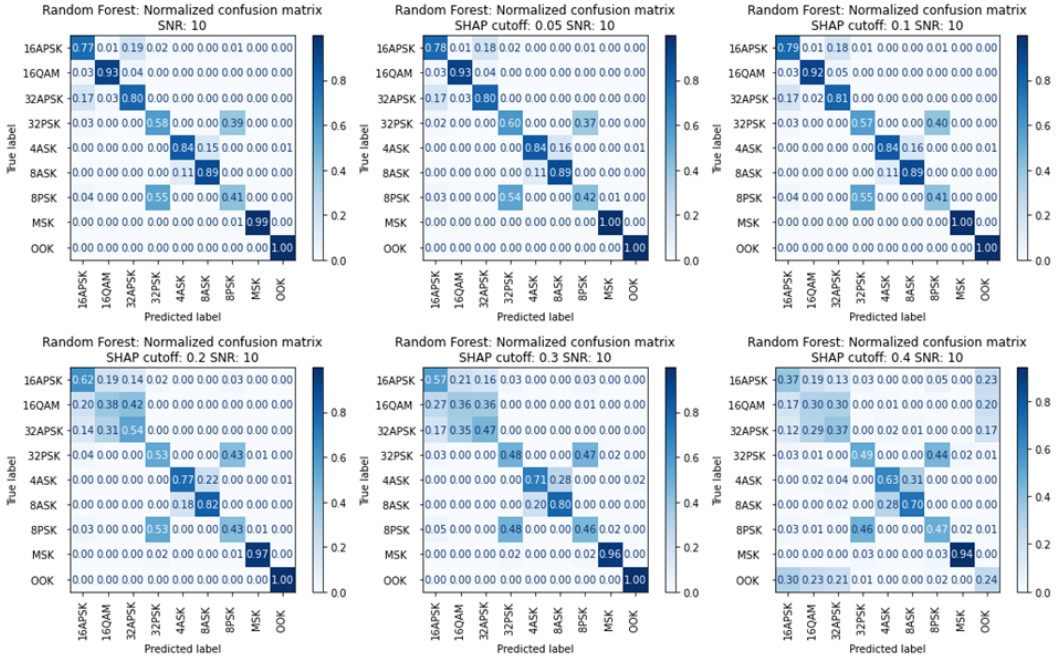

Figure 24: Random Forest Model at SNR = 10. Examining Random Forest prediction accuracy with confusion matrices. The baseline is given in the upper left-hand corner.

Random Forest SNR=10 OS=2				
Lowest SHAP value of Remaining Variables	Variables removed	Accuracy percent with all Variables	Accuracy percentage After Using SHAP to Remove Predictors	Percentage Lost
.05	1	80.069	80.528	-0.458
.1	5		80.333	-0.264
0.2	13		67.208	12.861
0.3	16		64.347	15.722
0.4	17		49.722	30.347

Figure 25: Random Forest Model at SNR = 10. Overall Random Forest prediction Accuracy on reduced cyclostationary feature set

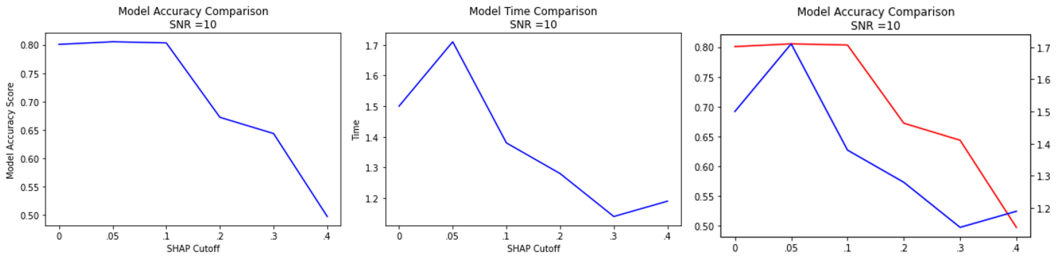

Figure 26: Random Forest Model at SNR = 10. Model accuracy vs. computational time. Excluding the 5 least significant cyclostationary features from the dataset could produce an optimal trade-off between model accuracy and processing time.